Recent Developments in Statistical Dialogue Systems

Steve Young

Machine Intelligence Laboratory
Information Engineering Division
Cambridge University Engineering Department
Cambridge, UK
Contents

- Review of Basic Ideas and Current Limitations
- Semantic Decoding
- Fast Learning
- Parameter Optimisation and Structure Learning
Recent Developments in Statistical Dialogue Systems, Braunschweig, Sept 2012 © Steve Young

Spoken Dialog Systems (SDS)

I want to find a restaurant?

inform(venuetype=restaurant)

What kind of food would you like?

request(food)

User

Recognizer

Semantic Decoder

Dialog Control

Database

Synthesizer

Message Generator

Dialog Acts

Waveforms

Words
A Statistical Spoken Dialogue System

Recent Developments in Statistical Dialogue Systems, Braunschweig, Sept 2012 © Steve Young

Supervised Learning

Ontology

Bayesian Belief Network

Evidence

Belief State

Response Generator

ASR

Inform(food=italian){0.6}
Inform(food=indian){0.2}
Inform(area=east){0.1}
null(){0.1}

Decision

Stochastic Policy

confirm(food=italian)
request(area)

Rewards: success/fail

Reward Function

Partially Observable Markov Decision Process (POMDP)

I want an Italian

Id like Italian{0.2}
Id like Indian{0.2}
In the east{0.1}

Your looking for an Italian restaurant, whereabouts?

Id like italian {0.4}
I want an Italian {0.2}
Id like Indian{0.2}
In the east{0.1}
The POMDP SDS Framework

\[ o_t = p(u_t | x_t) = \sum_{w_t} p(u_t | w_t) p(w_t | x_t) \]

\[ b_t(s_t) = \eta p(o_t | s_t, a_{t-1}) \sum_{s_{t-1}} P(s_t | s_{t-1}, a_{t-1}) b_{t-1}(s_{t-1}) \]

\[ a_t \sim \pi(b_t, a_t) = \frac{e^{\theta \cdot \phi_{a_t}(b_t)}}{\sum_a e^{\theta \cdot \phi_a(b_t)}} \]

\[ R = E \left\{ \sum_{s_t} r(s_t, a_t) b_t(s_t) \right\} \]

Maximise \( \theta \)
J. Williams (2007). "POMDPs for Spoken Dialog Systems." Computer Speech and Language 21(2)
Belief Monitoring (Tracking)

- **Inform**: inform(type=bar, food=french) {0.6}
- **Inform**: inform(type=restaurant, food=french) {0.3}
- **Confirm**: confirm(type=restaurant, food=french)
- **Affirm**: affirm() {0.9}
Choosing the next action – the Policy

Policy Vector

\[ \phi_1 \phi_2 \phi_3 \]

Quantize

<table>
<thead>
<tr>
<th>type</th>
<th>food</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Inform(type=bar) {0.4}

All Possible Summary Actions: inform, select, confirm, etc

Sample

Map

select(type=bar, type=restaurant)
Let’s Go 2010 Control Test Results

All Baseline Systems

Success Rate

Word Error Rate (WER)

Cambridge
89% Success
33% WER

Baseline
65% Success
42% WER

Another
75% Success
34% WER

B. Thomson
"Bayesian Update of State for the Let’s Go Spoken Dialogue Challenge."
SLT 2010.
Demo of Cambridge Restaurant Information

Call the system by pressing the call button to the right.
Issues with the 2010 System Design

- Poor coverage of N-best of semantic hypotheses
- Hand-crafting of summary belief space
- Slow policy optimisation and reliance on user simulation
- Dependence on hand-crafted dialogue model parameters
- Dependence on static ontology/database
N-best Semantic Decoding

Conventional N-best decoding

speech → ASR → N-best word strings → Semantic Decoder (applied N times) → N-best dialogue acts → Merge Duplicates → M-best dialogue acts
M<<N

Confusion Network decoding

speech → ASR → Word confusion network → Feature Extraction → n-gram features → Semantic Decoder (applied once) → M-best dialogue acts → M<<N
optional context
Confusion Network Decoder (Mairesse/Henderson)

$x_i = E\{C(n\text{-gram}_i)\}^{1/|n\text{-gram}_i|}$

$n = 1, 2, 3$

Context

Binary Vector

$b_j$ if $item_j$ in last system act
Confusion Network Decoder Evaluation

Comparison of item retrieval on corpus of 4.8k utterances
N-best hand-crafted Phoenix decoder vs Confusion network decoder (trained on 10k utterances)

Live Dialogue System

<table>
<thead>
<tr>
<th></th>
<th>N-best Phoenix</th>
<th>Confusion Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-score</td>
<td>0.80</td>
<td>0.82</td>
</tr>
<tr>
<td>ICE</td>
<td>2.02</td>
<td>1.26</td>
</tr>
<tr>
<td>Average Reward</td>
<td>10.6</td>
<td>11.15</td>
</tr>
</tbody>
</table>

Policy Optimization

Policy parameters chosen to maximize expected reward

\[ J(\theta) = E\left[ \frac{1}{T} \sum_t r(s_t, a_t) | \pi_{\theta} \right] \]

Natural gradient ascent works well

\[ \tilde{\nabla} J(\theta) = F^{-1}_\theta \nabla J(\theta) \]

Gradient is estimated by sampling dialogues and in practice Fisher Information Matrix does not need to be explicitly computed. This is the Natural Actor Critic Algorithm.

However,

A) slow (~100k dialogues) and
B) requires summary space approximations

Q-functions and the SARSA algorithm

Traditional reinforcement learning is commonly based on finding the optimal Q function:

$$Q^*(b, a) = \max_\pi \left[ E_\pi \left\{ \sum_{\tau=t+1}^{T} r(b_\tau, a_\tau) \right\} \right]$$

The optimal deterministic policy is then simply

$$\pi^*(b) = \arg\max_a [Q^*(b, a)]$$

$Q^*$ can be found sequentially using the SARSA algorithm

- $b = b_0$; choose action $a$ e-greedily from $\pi(b)$
- For each dialogue turn
  - Take action $a$, observe reward $r$ and next state $b'$
  - Choose action $a'$ e-greedily from $\pi(b')$
  - $Q(b, a) = Q(b, a) + \lambda [Q(b', a') - (Q(b, a) - r)]$
- $b = b'$; $a = a'$

Eventually, $Q \to Q^*$
For POMDPs, the belief space is continuous and direct representations of Q are intractable. However, Q can be approximated as a zero mean Gaussian process by designing a *kernel* to represent the correlations between points in belief x action space. Thus:

\[ Q(b,a) \sim \mathcal{GP}(0, k((b,a),(b,a))) \]

Given a sequence of state-action pairs

\[ B_t = [(b_0,a_0),(b_1,a_1),\ldots,(b_t,a_t)]' \]

and rewards

\[ r_t = [r_0,r_1,\ldots,r_t]' \]

there is a closed form solution for the posterior:

\[ Q(b,a) | B_t,r_t \sim N(\bar{Q}(b,a),\text{cov}((b,a),(b,a))) \]

This suggests a SARSA-like sequential optimisation:

- \( b = b_0 \); choose action a e-greedily from \( \bar{Q}(b,a) \)
- For each dialogue turn
  - Take action a, observe reward r and next state b’
  - choose action a’ e-greedily from \( \bar{Q}(b’,a’) \)
  - Update the posterior covariance estimate
- \( b = b’ \); a = a’
Benefits of GP-SARSA

- sequential estimation of distribution of Q (not Q itself)
- each new data point can impact on whole distribution via the covariance function → very efficient use of training data
- much faster learning than gradient methods such a Natural Actor Critic (NAC)

\[ \alpha = a_r g_m a_Q(b,a) \]

\[ \epsilon \text{-greedy exploration} \]

\[ a = \begin{cases} \arg\max_a \hat{Q}(b,a) \text{ with prob } 1 - \epsilon \\ \text{random action} \text{ with prob } \epsilon \end{cases} \]
Benefits of GP-SARSA

- variance of Q is known at each stage → more intelligent exploration:

Variance exploration

\[
\begin{align*}
\arg \max_a \left[ Q(b,a) \right] & \quad \text{with prob } 1 - \varepsilon \\
\arg \max_a \left[ \text{cov}((b,a),(b,a)) \right] & \quad \text{with prob } \varepsilon 
\end{align*}
\]

Stochastic policy

\[
Q^i(b,a_i) \sim N(\tilde{Q}(b,a_i), \text{cov}(b,a_i),(b,a_i))
\]

\[
a = \arg \max_{a_i} [Q^i(b,a_i)]
\]

Well trained within 3k dialogues
And summary space mapping no longer needed

“Gaussian processes for policy optimisation of large scale POMDP-based SDS”, Gasic et al, SLT 2012.
Parameter Estimation – Blaise Thomson

Goal

$g_t$

Transition Model

$p(g_{t+1}|g_t, a_t)$

User Behaviour

$p(u_t|g_t, a_{t-1})$

Recognition Errors

$p(o_t|u_t)$

Memory

$p(h_t|u_t, h_{t-1}, a_{t-1})$

These parameters typically hand-crafted. How can we learn them from data?

User Act

$u_t$

History

$h_t$

Observation at time $t$

$o_t$

$a_{t-1}$
Factor Graphs and Expectation Propagation

\[ f_1(S_1) \text{ where } S_1 = (a_{t-1}, g_t, u_t) \]

\[ b(s_t | o_t, s_{t-1}, a_{t-1}) \propto \prod_{k=1}^{K} f_k(S_k) \]

- exact computation is intractable
- can be approximated using belief propagation
- we use Expectation Propagation (EP)
- using EP factors can be discrete & continuous
- hence, parameters can be added and updated simultaneously
Effect of Parameter Learning on TownInfo System

![Graph showing success rate and its 95% confidence interval](image)

- **Learned Parameters**
- **Hand-crafted Parameters**

Success rate and its 95% confidence interval

Confusion rate (%)

Success rate

0 10 20 30 40 50 60

86 88 90 92 94 96 98 100
The ability to learn parameters can be extended to learn structure.

Let $G = \{ g_k \}$ be a set of Bayesian Networks (or Factor Graphs):

- Compute Evidence $p(D|g_k)$ for all $g_k$ in $G$
- Augment $G$ by perturbing each $g_k$
- Prune $G$
- Stop?
- Select $g_k$ with highest Evidence

Potential to learn:
- a) additional values for an existing variable
- b) new hidden variables
- c) new links between variables
Conclusions

- Statistical Dialogue Systems based on POMDPs are viable, offer increased robustness to noise and require no hand-crafting

- Good progress is being made on increasing accuracy and speeding up learning

- Learning directly on human users rather than depending on user simulators is now possible

- Current systems are built from static ontologies for closed domains

- Next steps will include building more flexible systems capable of dynamically adapting to new information content.

Acknowledgement - this work is the result of a team effort: Catherin Breslin, Milica Gasic, Matt Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis and former members of the CUED Dialogue Systems Group